

Software–Projekt 2 2013

VAK 03-BA-901.02

Testplan

IT_R3V0LUT10N

Sebastian Bredehöft	sbrede@tzi.de	2751589
Patrick Damrow	damsen@tzi.de	2056170
Tobias Dellert	tode@tzi.de	2936941
Tim Ellhoff	tellhoff@tzi.de	2520913
Daniel Pupat	dpupat@tzi.de	2703053

Inhaltsverzeichnis

1	Einführung (Sebastian)	3
1.1	Zweck	3
1.2	Umfang	3
1.3	Beziehungen zu anderen Dokumenten	3
1.4	Aufbau der Testbezeichner	3
1.5	Dokumentation der Testergebnisse	4
1.6	Definitionen und Akronyme	4
1.7	Referenzen	4
2	Systemüberblick (Sebastian)	4
2.1	Module der Anwendungsschicht und deren Funktionen	5
3	Merkmale (Daniel)	6
3.0.1	Funktionale Anforderungen	7
3.1	Nicht zu testende Merkmale	7
4	Abnahme- und Testendekriterien(Daniel & Sebastian)	7
5	Vorgehensweise (Sebastian)	8
5.1	Komponenten- und Integrationstest	8
5.2	Funktionstest	10
6	Aufhebung und Wiederaufnahme (Daniel)	10
7	Hardware- und Softwareanforderungen (Daniel)	11
7.1	Hardware	11
7.2	Software	11
8	Testfälle (Daniel & Sebastian)	11
8.1	Komponententest	11
8.2	Integrationstest	13
8.3	Funktionstest	15
8.4	Leistungstest	22
8.4.1	Härtetest	22
8.4.2	Volumentest	22
8.4.3	Sicherheitstest	22
8.4.4	Erholungstest	23
9	Testzeitplan (Daniel & Sebastian)	23

Version und Änderungsgeschichte

Version	Datum	Änderungen
1.0	18.12.2013	Dokumentvorlage als initiale Fassung kopiert
1.1	21.12.2013	Bis auf Testfälle und Testzeitplan vervollständigt
1.2	22.12.2013	Testplan abgabefertig

1 Einführung (Sebastian)

1.1 Zweck

Der Testplan bietet einen Überblick über die geplanten Tests und dient u.a. als Anleitung für die Tester. Die Software soll dabei ausführlich auf Funktionalität getestet werden.

Im Testplan wird festgelegt, wie man welche Komponenten testet. Dazu wird außerdem definiert, welchen Umfang die Tests haben sollen und wann ein Test erfolgreich ist und wann nicht.

Während der Implementierungsphase werden wir uns nach dem Testplan richten und ihn gegebenenfalls weiterführen und vervollständigen.

1.2 Umfang

Der Testplan entspricht der vereinfachten Form des *IEEE Standard for Software Test Documentation 829-1998*.

1.3 Beziehungen zu anderen Dokumenten

Dieser Testplan bezieht sich zum einen auf die Anforderungsspezifikation, da dort die Systemeigenschaften und Systemattribute spezifiziert wurden. Die Testfälle werden auf Grundlage der dortigen Anwendungsfälle entwickelt.

Außerdem gibt es Referenzen zur Architekturbeschreibung, da in dieser die Module und Komponenten definiert wurden, die in diesem Dokument getestet werden sollen.

1.4 Aufbau der Testbezeichner

Der Aufbau der Testbezeichner richtet sich nach folgendem Schema:

- Die ersten beiden Buchstaben geben die Art des Tests vor. Dabei unterscheiden wir zwischen vier verschiedenen Testarten:
 - Komponententests = KT

- Integrationstests = IT
- Funktionstests = FT
- Leistungstests = LT
- Die Nummer steht für die jeweilige Testfallnummer
- Optional: in alphabetischer Reihenfolge werden hier Variationen oder untergeordnete Testfälle definiert

Nach diesem Schema sieht ein Testbezeichner nun folgendermaßen aus:

IT-3-A: Integrationstest, Nr. 3, Variante 1

1.5 Dokumentation der Testergebnisse

Zu jedem Testfall wird ein kurzes Testprotokoll angefertigt. Dieses beinhaltet den Ablauf des Testfalls und die möglichen Komplikationen, die bei der Durchführung entstehen können. Dann werden die Resultate des Testfalls bestimmt und eventuell gefundene Fehler beschrieben.

1.6 Definitionen und Akronyme

1.7 Referenzen

IEEE Standard for Software Test Documentation 829-1998

<http://standards.ieee.org/findstds/standard/829-1998.html>

2 Systemüberblick (Sebastian)

Das System besteht aus der Server- und der Clientkomponente. Die konzeptionelle Sicht der Architekturbeschreibung (vgl. Abschnitt 3 der Architekturbeschreibung) dient als Grundlage für den Testplan, da dort die verschiedenen Komponenten beschrieben werden.

Auf der Serverseite gibt es die Komponenten **Communication**, **BusinessLogic** und **Persistence** (vgl. Abbildung 3: Konzeptionelle Sicht Server; Architekturbeschreibung).

Die Clientseite besteht aus den Komponenten **Communication**, **Model** und **User Interface** (vgl. Abbildung 4: Konzeptionelle Sicht Client; Architekturbeschreibung).

Da starke Abhängigkeiten zwischen all diesen Komponenten bestehen, ist es wichtig, dass diese Komponenten fehlerfrei funktionieren.

2.1 Module der Anwendungsschicht und deren Funktionen

In der nachfolgenden Tabelle werden die Module verfeinert die in Punkt 5 visualisiert sind.

GUI	webapp(xhtml)
AndroidApp	AsyncBookTask.java BookAdapter.java MainActivity.java Show-BookActivity.java
Model	it_r3v.bibjsf.presentation
Communication	Network.java
UserInterface	GUI AndroidApp
BusinessLogic	AdministrationHandler.java MediumHandler.java BusinessHandler.java BusinessObjectHandler.java ReaderHandler.java LibrarianHandler.java BorrowHandler.java
BibCommon	Medium.java BusinessObject.java Reader.java Admin.java Librarian.java
Persistence	Data.java Persistence.java

3 Merkmale (Daniel)

Zu testende Merkmale sind in erster Linie Funktionen, die in den Mindestanforderungen enthalten sind. Dabei ist zu beachten, dass die Funktionen des Lesers für die Website und der App getestet werden müssen, während sich die anderen Testmerkmale auf die Website beziehen. Beide sind im Folgenden aufgelistet:

1. Benutzerrechte

- 1.1 Administrator
- 1.2 Bibliothekar
- 1.3 Leser
- 1.4 Gast

2. Administrator

- 2.1 Bibliothekar hinzufügen
- 2.2 Bibliothekar löschen
- 2.3 Bibliothekar ändern

3. Bibliothekar

- 3.1 Medium hinzufügen
- 3.2 Medium löschen
- 3.3 Medium ändern
- 3.4 Medium ausleihen
- 3.5 Medium zurücknehmen
- 3.6 Abgabedaten und Mahngebühren ändern
- 3.7 Vormerkungen anzeigen
- 3.8 Übersicht über verliehene Bücher(Versäumnisse, Mahngebühren)
- 3.9 Statistiken anzeigen

4. Leser

- 4.1 Medium suchen
- 4.2 Medium anzeigen

4.3 Medium vormerken

Dazu kommt noch der Gast, der unangemeldet nur suchen kann und man sollte die Unterklassen von Medium noch einzeln testen, ob diese die geforderten Attribute und Funktionen enthalten.

3.0.1 Funktionale Anforderungen

Besonders wichtig sind die Funktionen des Bibliothekars (3.1-3.5). Die sollten gut getestet werden, da es beim Ausleihvorgang nicht zu Problemen kommen soll, sodass irgendwo Bücher verschwinden oder Ausleiher oder Bücher verwechselt werden. Auch wichtig ist die Benutzerunterscheidung, damit Leser nicht irgendwas löschen oder hinzufügen. Außerdem sollte man die Suche und das Anzeigen der Bücher ausgiebig testen, da dies die Hauptfunktionen des Lesers sind.

3.1 Nicht zu testende Merkmale

Nicht zu testende Merkmale sind in erster Linie alle trivialen Funktionen. Zudem brauchen bereits implementierte Funktionen, wie `Leser verwalten`, `Backup` und `Restore`, `Buchaufkleber drucken`, `Leserausweise drucken`, `Import` und `Export von Buch- und Leserdaten` nicht mehr getestet werden, sofern man diese nicht verändert. Da wir `Buch verwalten` noch verändern, da wir mit mehreren Medientypen arbeiten, ist dieses wie in 3 beschrieben noch zu testen.

4 Abnahme- und Testendekriterien(Daniel & Sebastian)

Fehler werden in eine Kategorie eingeordnet und erhalten entsprechende Fehlerwerte. Aus diesen Fehlerwerten ergeben sich Prioritäten, die die Reihenfolge der Fehlerbehandlung angibt. Das Testen wird beendet, wenn der berechnete Fehlerwert aller Fehler pro 1000 Zeilen Code unter dem Wert 10 liegt und die Software nicht beeinträchtigt wird, d.h. es keinen Fehler der Fehlerklasse `Mittel` oder höher gibt.

Testabdeckung Die Testabdeckung soll so hoch wie möglich sein. Für ein stabiles System spricht, das die Testabdeckung in systemkritischen Bereichen soweit vollständig ist. Jeder Fehler in diesem Bereich kann das System zum Absturz bringen und muss somit verhindert werden. In anderen Bereichen die das laufende System bei einem Fehler weniger beeinträchtigen wird die Testabdeckung nicht so vollständig sein, wie in kritischen Bereichen.

Fehlerbewertung:

Die nachfolgende Tabelle spezifiziert die Auswirkung eines Fehlers, durch die man diese nach Priorität einordnen kann.

Fehlerkl. ¹	Beschreibung	Wert
Leicht	Unwesentliche Fehler, die den Programmablauf nicht beeinträchtigen, aber trotzdem behandelt werden sollten.	1
Mittel	Fehler in dieser Art haben Auswirkungen auf den Programmablauf. Dieser beeinträchtigt aber nicht die grundlegenden Funktionen.	10
Schwer	Fehler der Klasse „Schwer“ beeinträchtigen die Funktionsfähigkeit des Systems sehr stark und müssen sofort behandelt werden.	20
Fatal	Diese Fehler machen den Programmablauf unmöglich und können zum Absturz des Systems führen.	100

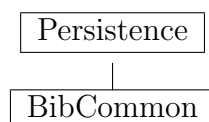
5 Vorgehensweise (Sebastian)

5.1 Komponenten- und Integrationstest

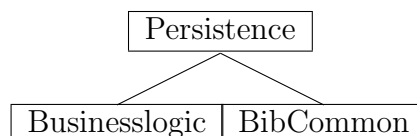
Die beiden Komponenten Server und Client werden bezüglich der Integrationstests zunächst unabhängig voneinander getestet und erst bei wenn das sichere Laufen der einzelnen Komponente gewährleistet ist, wird das System als ganzes getestet.

Server

Es wird zuerst die Persistenz mit **BibCommon** und **Persistence** getestet:

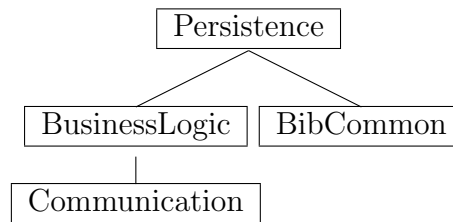


Nun folgt das Zusammenspiel mit der BusinessLogic **BusinessLogic**:



Darauf folgt nun die Kommunikation **Communication** mit den vorigen Komponenten:

¹=Fehlerklasse

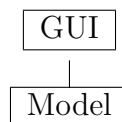


Somit ist der Server als ganzes zu testen, da jede einzelnen Komponenten mit ihren jeweiligen Abhängigkeiten getestet wurden.

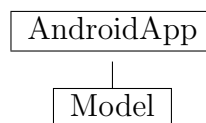
Client

Hier sind nun zwei verschiedene Komponenten für die Darstellung auf dem jeweiligen Endgerät vorhanden:

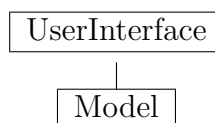
Einmal GUI welche zusammen mit dem Model getestet wird und sich an die Browserdarstellung richtet:



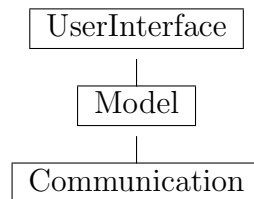
Und einmal Android App mit dem Model, welche sich an mobile Geräte richtet:



Diese beiden Tests werden zusammengefasst zu **User Interface** und zusammen mit der nächsten Ebene, dem Model getestet:



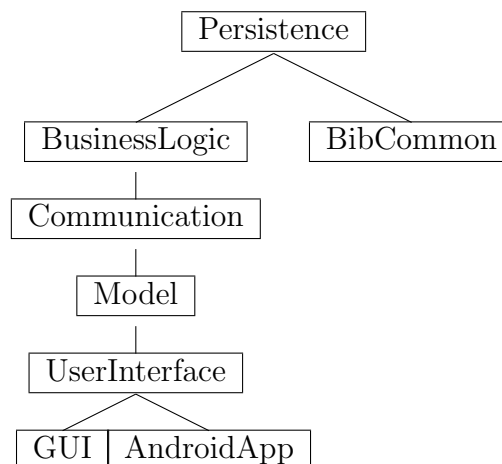
Darauf folgt nun der Test mit der Komponente **Communication**:



Nun ist das Zusammenspiel der Komponenten des Clients gewährleistet.

Server + Client

Um die kompletten Systemkomponenten zu testen werden jetzt der komplette Client und der komplette Server zusammen getestet:



5.2 Funktionstest

Die Funktionstest sind durch jene Anwendungsfälle aus der Anforderungsspezifikation vorgegeben. Jede dieser Funktionen muss durch Tests gedeckt sein.

6 Aufhebung und Wiederaufnahme (Daniel)

Wir werden Tests unterbrechen, wenn ein gewisser Wert überschritten wird, welcher über die Tabelle in Abschnitt 4 berechnet wird. In diesem Fall werden wir sofort wieder mit der Implementierung anfangen. Da wir mit der Bottom-up Strategie testen, werden wir bei Fehlern in der unteren Schicht einen niedrigeren Wert nehmen.

Bei Fehlern der Data setzen wir einen Wert von 10, bei Fehlern in der Logik einen Wert von 20 und bei Fehlern, welche die GUI betreffen, einen von 40 und bei den restlichen Faktoren einen von 100.

Sollten die Fehler behoben sein, testen wir noch einmal alle Komponenten, die mit den veränderten interagieren.

7 Hardware- und Softwareanforderungen (Daniel)

7.1 Hardware

Als Hardware stehen uns unsere Notebooks und Smartphones, sowie die Unirechner zur Verfügung. Dabei haben wir alle geforderten Betriebssysteme mindestens einmal auf unseren Notebooks installiert, sodass wir auf jeden Gerät testen können. Wir besitzen ebenfalls einen PC, der über Windows 2000 läuft, darüber testen wir auch noch, da dies den Rechnern des Kunden entspricht. Da nur Android Unterstützung gefordert ist, werden wir die App über unseren vorhandenen Smartphones, die Android haben, testen.

7.2 Software

Als Software benutzen wir in der Eclipse Umgebung JUnit-Tests. Diese werden in Form von BlackBox- und WhiteBox-Tests implementiert. Die App werden wir mithilfe eines Android Emulators testen.

8 Testfälle (Daniel & Sebastian)

8.1 Komponententest

Wir haben hier alle Klassen aufgelistet, welche wir testen wollen. Dabei werden wir keine abstrakten Klassen und Interfaces testen. Vorgegebene Klassen, welche bereits funktionieren, werden wir nicht testen. Exceptions testen wir nicht einzeln, sondern diese werden mit den zugehörigen Methoden getestet.

Klasse	Implementierer	Tester	Testart
AsyncBookTask	Patrick	Tim	Blackbox
BookAdapter	Patrick	Daniel	Blackbox
MainActivity	Patrick	Sebastian	Blackbox
Network	Patrick	Tobias	Blackbox
ShowBookActivity	Patrick	Tim	Blackbox
Admin	Tim	Daniel	Blackbox
Medium	Tim	Patrick	Blackbox
Reader	Tim	Tobias	Blackbox
Librarian	Tim	Daniel	Blackbox
Config	Tim	Sebastian	Blackbox
AdministrationHandler	Daniel	Patrick	Blackbox
BorrowHandler	Daniel	Tim	Blackbox
MediumHandler	Daniel	Tobias	Blackbox
LibrarianHandler	Daniel	Tim	Blackbox
ReaderHandler	Daniel	Tim	Blackbox
Data	Daniel	Tobias	Blackbox
AddLibrarianForm	Tobias	Patrick	Blackbox
AddMediumForm	Tobias	Tim	Blackbox
AddReaderForm	Tobias	Sebastian	Blackbox
Administration	Tobias	Patrick	Blackbox
AuthBackingBean	Tobias	Tim	Blackbox
MediumListDataModel	Sebastian	Patrick	Blackbox
MediumTable	Sebastian	Daniel	Blackbox
ReaderTable	Sebastian	Tobias	Blackbox
ChangeMediumForm	Sebastian	Tim	Blackbox
ChangeReaderForm	Sebastian	Daniel	Blackbox
TableDataModel	Sebastian	Patrick	Blackbox
BibServices	Sebastian	Tim	Blackbox

Tabelle 1: Komponententests

8.2 Integrationstest

Testfallbezeichner	IT-1-a Medium hinzufügen
Testobjekte	Persistence, Bibcommon
Eingabe	Mediendaten(Titel, Autor etc.)
Ausgabe	Erfolgreich Medium hinzugefügt
Umgebungserfordernisse	Glassfishserver läuft, Datenbank existiert
Anforderungen	keine
Abhängigkeiten	keine
Testfallbezeichner	IT-1-b Medium löschen
Testobjekte	Persistence, Bibcommon
Eingabe	MediumID
Ausgabe	Erfolgreich Medium gelöscht
Umgebungserfordernisse	Glassfishserver läuft, Datenbank existiert
Anforderungen	Medium ist in Datenbank vorhanden
Abhängigkeiten	IT-1-a
Testfallbezeichner	IT-2-a Leser hinzufügen
Testobjekte	BusinessLogic, Persistence, Bibcommon
Eingabe	Leserdaten(Name, Nachname etc.)
Ausgabe	Erfolgreich Leser hinzugefügt
Umgebungserfordernisse	Glassfishserver läuft, Datenbank existiert
Anforderungen	keine
Abhängigkeiten	keine
Testfallbezeichner	IT-2-b Leser löschen
Testobjekte	BusinessLogic, Persistence, Bibcommon
Eingabe	ReaderID
Ausgabe	Erfolgreich Leser gelöscht
Umgebungserfordernisse	Glassfishserver läuft, Datenbank existiert
Anforderungen	Reader ist in Datenbank vorhanden
Abhängigkeiten	IT-2-a
Testfallbezeichner	IT-3-a Alle Medien zurückgeben
Testobjekte	Communication, BusinessLogic, Persistence, Bibcommon
Eingabe	Mediendatenbank
Ausgabe	Liste zurückgeben
Umgebungserfordernisse	Glassfishserver läuft, Datenbank existiert
Anforderungen	Datenbank enthält verschiedene Medien
Abhängigkeiten	keine

Testfallbezeichner	IT-3-b Alle Bücher zurückgeben
Testobjekte	Communication, BusinesLogic, Persistence, Bibcommon
Eingabe	Medienart, nach welcher sortiert werden soll
Ausgabe	Liste zurückgeben
Umgebungserfordernisse	Glassfishserver läuft, Datenbank existiert
Anforderungen	Datenbank enthält verschiedene Medien
Abhängigkeiten	keine

Testfallbezeichner	IT-4-a Alle Medien als Liste anzeigen
Testobjekte	GUI, Model
Eingabe	Liste anzeigen
Ausgabe	Medienliste anzeigen
Umgebungserfordernisse	Website ist aufgerufen, Glassfishserver läuft, Datenbank existiert
Anforderungen	Datenbank enthält verschiedene Medien
Abhängigkeiten	IT-1 bis IT-3

Testfallbezeichner	IT-4-b Alle Medien als Liste anzeigen
Testobjekte	AndroidApp, Model
Eingabe	Liste anzeigen
Ausgabe	Medienliste anzeigen
Umgebungserfordernisse	App ist gestartet, Glassfishserver läuft, Datenbank existiert
Anforderungen	Datenbank enthält verschiedene Medien
Abhängigkeiten	IT-1 bis IT-3

Testfallbezeichner	IT-5 Angezeigte Medienliste nach Titel sortieren
Testobjekte	UserInterface, Model
Eingabe	Titel sortieren
Ausgabe	sortierte Liste
Umgebungserfordernisse	Website ist aufgerufen/App ist gestartet, Glassfishserver läuft, Datenbank existiert
Anforderungen	Datenbank enthält verschiedene Medien
Abhängigkeiten	IT-1 bis IT-3

Testfallbezeichner	IT-6 Dvd suchen
Testobjekte	Communication, UserInterface, Model
Eingabe	Suchbegriff
Ausgabe	gefundene Dvd
Umgebungserfordernisse	Website ist aufgerufen/App ist gestartet, Glassfishserver läuft, Datenbank existiert
Anforderungen	Datenbank enthält verschiedene Medien
Abhängigkeiten	IT-1 bis IT-3

Testfallbezeichner	IT-7 Ausleihhistorie eines Benutzers anzeigen
Testobjekte	Communication, UserInterface, Model, BusinessLogic, Persistence, BibCommon
Eingabe	Eine BenutzerID
Ausgabe	Liste der ausgeliehenen Medien
Umgebungserfordernisse	Website ist aufgerufen, Glassfishserver läuft, Datenbank existiert,
Anforderungen	Benutzer ist Bibliothekar, gesuchter Benutzer befindet sich in Datenbank, Datenbank enthält verschiedene Medien
Abhängigkeiten	IT-1 bis IT-6

Testfallbezeichner	IT-8 Einloggen
Testobjekte	Communication, UserInterface, Model, BusinessLogic, Persistence, BibCommon
Eingabe	Logindaten
Ausgabe	erfolgreich eingeloggt
Umgebungserfordernisse	Website ist aufgerufen/App ist gestartet, Glassfishserver läuft, Datenbank existiert
Anforderungen	Nutzer existiert, Datenbank enthält verschiedene Medien
Abhängigkeiten	IT-1 bis IT-6

8.3 Funktionstest

Bezeichner:	FT-1
Anwendungsfall	Login
Eingabe	Anmeldedaten(Nutzername, Passwort)
Ausgabe	Rückmeldung über erfolgreiche Anmeldung Bei Fehlerfall: Rückmeldung Anmeldung nicht erfolgreich
Umsetzung	Manuelle Eingabe der Nutzerdaten, Benutzer ist registriert

Bezeichner:	FT-2
Anwendungsfall	Logout
Eingabe	Logout-Button wird gedrückt
Ausgabe	Rückmeldung über erfolgreiche Abmeldung Bei Fehlerfall: Rückmeldung Abmeldung nicht erfolgreich
Umsetzung	Manuelles ausloggen, Benutzer ist registriert

Bezeichner:	FT-3
Anwendungsfall	Leserinformationen anzeigen
Eingabe	Detailbutton in der Leserliste wird gedrückt Alternativ: Lesersuche und dann auf Leser klicken
Ausgabe	Leserinformation Bei Fehlerfall: Fehlerrückmeldung
Umsetzung	Manuell auf einen Leser klicken; Benutzer ist Bibliothekar

Bezeichner:	FT-4
Anwendungsfall	Vormerkung bearbeiten
Eingabe	Benutzer sieht seine Vormerkungen; Bearbeiten bei einer Vormerkung klicken
Ausgabe	Rückmeldung über erfolgreiches Ändern Bei Fehlerfall: Rückmeldung Ändern war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist registrierter Leser

Bezeichner:	FT-5
Anwendungsfall	Alle Medien anzeigen
Eingabe	Medienliste aufrufen
Ausgabe	Medienliste wird angezeigt Bei Fehlerfall: Medienliste wird nicht angezeigt
Umsetzung	Manuell, beliebiger Benutzer

Bezeichner:	FT-6
Anwendungsfall	Medien hinzufügen
Eingabe	Mediendaten (Manuell oder durch Scanner)
Ausgabe	Rückmeldung über erfolgreiches Hinzufügen Bei Fehlerfall: Rückmeldung Hinzufügen war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar

Bezeichner:	FT-7
Anwendungsfall	Dasselbe Medium wird hinzugefügt
Eingabe	Mediendaten eines schon vorhandenen Mediums
Ausgabe	Fehlermeldung; Medium schon vorhanden Bei Fehlerfall: Medium wurde erfolgreich hinzugefügt
Umsetzung	Manuell, Benutzer ist Bibliothekar

Bezeichner:	FT-8
Anwendungsfall	Das gleiche Medium wird hinzugefügt (anderes Exemplar)
Eingabe	Mediendaten eines schon vorhandenen Mediums, mit individueller ID
Ausgabe	Rückmeldung über erfolgreiches Hinzufügen des Exemplars Bei Fehlerfall: Rückmeldung Hinzufügen war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar

Bezeichner:	FT-9
Anwendungsfall	Medium bearbeiten
Eingabe	neue Mediendaten
Ausgabe	Rückmeldung über erfolgreiches Ändern Bei Fehlerfall: Rückmeldung Ändern war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar

Bezeichner:	FT-10
Anwendungsfall	Medium löschen
Eingabe	MediumID
Ausgabe	Rückmeldung über erfolgreiches Löschen Bei Fehlerfall: Rückmeldung Ändern war nicht erfolgreich
Umsetzung	automatisch

Bezeichner:	FT-11
Anwendungsfall	CSV-Import
Eingabe	CSV-Datei
Ausgabe	Rückmeldung über erfolgreiches Importieren Bei Fehlerfall: Rückmeldung Importieren war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar

Bezeichner:	FT-12
Anwendungsfall	CSV-Export
Eingabe	Datenbank
Ausgabe	CSV-Datei der Eingabe Bei Fehlerfall: Rückmeldung Exportieren war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar

Bezeichner:	FT-13
Anwendungsfall	Medium suchen
Eingabe	Suchbegriff
Ausgabe	Liste von gefundenen Medien Bei Fehlerfall: Rückmeldung Suche war nicht erfolgreich
Umsetzung	Manuell, beliebiger Nutzer

Bezeichner:	FT-14
Anwendungsfall	Medium anzeigen
Eingabe	Auf ein Medium in einer Liste klicken
Ausgabe	Mediendetails werden angezeigt Bei Fehlerfall: Details werden nicht angezeigt
Umsetzung	Manuell, beliebiger Nutzer

Bezeichner:	FT-15
Anwendungsfall	Medium ausleihen
Eingabe	Medium und Leser
Ausgabe	Rückmeldung über erfolgreiches Ausleihen Bei Fehlerfall: Rückmeldung Ausleihen war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar

Bezeichner:	FT-16
Anwendungsfall	Ausleihzeitraum anzeigen
Eingabe	Leser, der Medien ausgeliehen hat
Ausgabe	Ausleihzeitraum Bei Fehlerfall: Ausleihzeitraum wird nicht angezeigt
Umsetzung	Manuell, Benutzer ist Bibliothekar
Bezeichner:	FT-17
Anwendungsfall	Medium zurückgeben
Eingabe	Medium und Leser
Ausgabe	Rückmeldung über erfolgreiches zurückgeben Bei Fehlerfall: Rückmeldung Zurückgeben war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar
Bezeichner:	FT-18
Anwendungsfall	Medium vormerken
Eingabe	Medium
Ausgabe	Rückmeldung über erfolgreiches Vormerken Bei Fehlerfall: Rückmeldung Vormerken war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist registrierter Leser
Bezeichner:	FT-19
Anwendungsfall	Leserliste anzeigen
Eingabe	Leserliste aufrufen
Ausgabe	Leserliste Bei Fehlerfall: Leserliste wird nicht angezeigt
Umsetzung	Manuell, Benutzer ist Bibliothekar
Bezeichner:	FT-20
Anwendungsfall	Leser hinzufügen
Eingabe	Leserdaten
Ausgabe	Rückmeldung über erfolgreiches Hinzufügen Bei Fehlerfall: Rückmeldung Hinzufügen war nicht erfolgreich
Umsetzung	automatisch
Bezeichner:	FT-21
Anwendungsfall	Leser löschen
Eingabe	LeserID
Ausgabe	Rückmeldung über erfolgreiches Löschen Bei Fehlerfall: Rückmeldung Löschen war nicht erfolgreich
Umsetzung	automatisch

Bezeichner:	FT-22
Anwendungsfall	Leser bearbeiten
Eingabe	neue Leserdaten
Ausgabe	Rückmeldung über erfolgreiches Bearbeiten Bei Fehlerfall: Rückmeldung Bearbeiten war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar

Bezeichner:	FT-23
Anwendungsfall	Leser sperren
Eingabe	LeserID
Ausgabe	Rückmeldung über erfolgreiches Sperren Bei Fehlerfall: Rückmeldung Sperren war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar

Bezeichner:	FT-24
Anwendungsfall	Leser suchen
Eingabe	Suchbegriff
Ausgabe	Suchergebnis Bei Fehlerfall: Rückmeldung Suchen war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar

Bezeichner:	FT-25
Anwendungsfall	Bibliothekare anzeigen
Eingabe	BibliothekarID
Ausgabe	Bibliothekarliste Bei Fehlerfall: leere Liste
Umsetzung	Manuell, Benutzer ist Administrator

Bezeichner:	FT-26
Anwendungsfall	Bibliothekar löschen
Eingabe	BibliothekarID
Ausgabe	Rückmeldung über erfolgreiches Löschen Bei Fehlerfall: Rückmeldung Löschen war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Administrator

Bezeichner:	FT-27
Anwendungsfall	Bibliothekar bearbeiten
Eingabe	BibliothekarID
Ausgabe	Rückmeldung über erfolgreiches Bearbeiten Bei Fehlerfall: Rückmeldung Bearbeiten war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Administrator

Bezeichner:	FT-28
Anwendungsfall	Bibliothekar hinzufügen
Eingabe	Bibliothekardaten
Ausgabe	Rückmeldung über erfolgreiches Hinzufügen Bei Fehlerfall: Rückmeldung Hinzufügen war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Administrator
Bezeichner:	FT-29
Anwendungsfall	Statistiken anzeigen: am meisten ausgeliehene Bücher
Eingabe	Statistiken aufrufen
Ausgabe	Statistik wird angezeigt Bei Fehlerfall: Rückmeldung Anzeigen war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar
Bezeichner:	FT-30
Anwendungsfall	Statistiken anzeigen: am wenigsten ausgeliehene Bücher
Eingabe	Statistiken aufrufen
Ausgabe	Statistik wird angezeigt Bei Fehlerfall: Rückmeldung Anzeigen war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar
Bezeichner:	FT-31
Anwendungsfall	Mahnungsliste anzeigen
Eingabe	Mahnungsliste aufrufen
Ausgabe	Mahnungsliste wird angezeigt Bei Fehlerfall: Rückmeldung Anzeige war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar
Bezeichner:	FT-32
Anwendungsfall	Mahnungsliste drucken
Eingabe	Druckbutton drücken
Ausgabe	Drucken erfolgreich Bei Fehlerfall: Rückmeldung Drucken war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar
Bezeichner:	FT-33
Anwendungsfall	Startseite bearbeiten
Eingabe	Startseiteninhalt
Ausgabe	Überarbeitete Startseite Bei Fehlerfall: Rückmeldung Bearbeiten war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar

Bezeichner:	FT-34
Anwendungsfall	Abgabedaten/ Mahngebühren bearbeiten
Eingabe	neue Abgabedaten/ Mahngebühren
Ausgabe	Bearbeiten erfolgreich Bei Fehlerfall: Rückmeldung Bearbeiten war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar

Bezeichner:	FT-35
Anwendungsfall	Vormerkung anzeigen
Eingabe	BuchID
Ausgabe	Liste der Vormerkungen Bei Fehlerfall: Rückmeldung Anzeigen war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar

Bezeichner:	FT-36
Anwendungsfall	Zeige letzten Ausleiher
Eingabe	BuchID
Ausgabe	letzter Ausleiher Bei Fehlerfall: Rückmeldung Anzeigen war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist Bibliothekar

Bezeichner:	FT-37
Anwendungsfall	Liste ausgeliehener Bücher anzeigen
Eingabe	Historie anzeigen klicken
Ausgabe	Liste der ausgeliehenen Bücher Bei Fehlerfall: Rückmeldung Anzeigen war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist registrierter Benutzer

Bezeichner:	FT-38
Anwendungsfall	automatisches Backup
Eingabe	Backupintervall (Wann immer ein Backup durchgeführt werden soll)
Ausgabe	Backup erfolgreich Bei Fehlerfall: Rückmeldung Backup war nicht erfolgreich
Umsetzung	automatisch

Bezeichner:	FT-39
Anwendungsfall	Verlängerungswunsch einreichen
Eingabe	Button Verlängern klicken
Ausgabe	Verlängerungswunsch wird abgeschickt Bei Fehlerfall: Rückmeldung Wunsch einreichen war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist registrierter Leser

Bezeichner:	FT-40
Anwendungsfall	Ausleihhistorie einschalten
Eingabe	Einschalten klicken
Ausgabe	Ausleihhistorie aktiviert Bei Fehlerfall: Rückmeldung Aktivierung war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist registrierter Leser

Bezeichner:	FT-41
Anwendungsfall	Ausleihhistorie ausschalten
Eingabe	Ausschalten klicken
Ausgabe	Ausleihhistorie deaktiviert Bei Fehlerfall: Rückmeldung Deaktivierung war nicht erfolgreich
Umsetzung	Manuell, Benutzer ist registrierter Leser

8.4 Leistungstest

8.4.1 Härtetest

Bezeichner:	LT-1
Beschreibung	100 Nutzer lassen sich gleichzeitig die Bücherliste anzeigen
Ziel:	Robustheit der Datenbank mit vielen Anfragen umzugehen wird getestet
Bei Erfolg:	Datenbank kann Anfragen bearbeiten; es gibt keine langen Wartezeiten
Fehler:	Timeout, Absturz

8.4.2 Volumentest

Bezeichner:	LT-2
Beschreibung	Eine sehr große CSV-Datei wird importiert
Ziel:	Robustheit der Datenbank mit großen Datenmengen umzugehen wird getestet
Bei Erfolg:	Datenbank kann mit der Verarbeitung umgehen; es gibt Fehler
Fehler:	Timeout, Absturz

8.4.3 Sicherheitstest

Bezeichner:	LT-3
Beschreibung	Ein Nutzer gibt ein falsches Passwort ein
Ziel:	Korrekte Authentifizierung wird getestet
Bei Erfolg:	Nutzer kann sich nicht einloggen; System gibt Fehlermeldung
Fehler:	Benutzer kann sich anmelden

8.4.4 Erholungstest

Bezeichner:	LT-4
Beschreibung	Ein Nutzer gibt mehrmals ein falsches Passwort ein
Ziel:	Erholt sich das System; Kann man sich danach problemlos mit dem richtigen Passwort einloggen
Bei Erfolg:	Bei richtiger Eingabe der Logindaten, ist man eingeloggt
Fehler:	Man kann sich nicht mehr einloggen, Absturz

Bezeichner:	LT-5
Beschreibung	Der Server geht offline und startet die Verbindung neu
Ziel:	Ist das System nach neuer Verbindung wieder funktionstüchtig
Bei Erfolg:	System produziert keine Fehler
Fehler:	Funktionen werden nicht mehr unterstützt, Datenverlust, TimeOut, Absturz

9 Testzeitplan (Daniel & Sebastian)

Komponententests: Woche ab 20.1.2014

Integrationstests: Woche ab 20.1.2014

Funktionstests: Woche ab 20.1.2014

Komponententests v2: 5.2.2014 bis 10.2.2014

Leistungstests: 15.2.2014 bis 23.2.2014

Akzeptanztests: Woche vom 10.02.2014 bis 15.02.2014